Submitted to the Annals of Applied Statistics
arXiv: arXiv:0000.0000

ANALYZING HIGH-SPEED NETWORK DATA

By Kejia Hu[*,†], Jaesik Choi[†,‡] Alex Sim[†] and Jiming Jiang[*]

[*] *University of California, Davis*
[†] *Lawrence Berkeley National Laboratory*
[‡] *Ulsan National Institute of Science and Technology*

Analyzing network data is an important problem to network users and network designers to optimize network usage during operations or reorganize network structure. We present a new predictive approach to the analysis of the network performance in traffic patterns and variation with the network conditions via the best predictive generalized linear mixed model (GLMM). The GLMM is built for the best predictive performance, in which the parameter estimates are obtained by minimizing the mean-squared prediction error (MSPE). In order to deal with the big data collected by the network, the best predictive GLMM is combined with the Lasso to enhance the computation efficiency, and an innovative approach using the bootstrap is discussed. Both the network data and simulation studies support our new approach in that (1) the highest prediction accuracy even under a model misspecification; and (2) the least computation time compared to the Estimation-oriented GLMM with Lasso and Stepwise Selection GLMM. A major computational advantage of our method is that, unlike some of the current approaches, our method does not require the EM procedure.

1. Introduction. Efficient data access is essential for sharing massive amounts of data among many geographically distributed collaborators. The analysis of network traffic is getting more and more important today to efficiently utilize the limited resources offered by the network infrastructures and plan wisely large data transfers. Data transfer performance for large dataset can be improved by learning the current condition and accurately predicting the future network performance. Short-term prediction of network traffic performance guides the immediate scientific data placements for network users. Long-term forecast of network traffic enables the capacity planning of the network infrastructure up to the future needs for network designers. Such predictions become non-trivial when amounts of network measurement data grow in unprecedented speed and volumes, and misspecified models are used. The available data sources are SNMP [1] and flow-level

Keywords and phrases: Generalized Linear Mixed Models (GLMM), Mean squared prediction error (MSPE), Model misspecification, Lasso regularization, Tuning Parameters Selection

data such as Cisco's NetFlow[2]. SNMP provides low-volume data which is a single time series regarding the time and the corresponding traffic volume in the network. Based on SNMP, statistical network analysis has conducted recently by Hu et al [7] and Antoniades et al [8]. On the other hand, NetFlow measurements provide high volume with abundant specific information of each data flow such as time, path and delivery condition.

Based on the NetFlow measurements, statistical models are built to predict network usage in this paper. For network users, an accurate prediction of duration of a transfer can help choosing the start time, the path and the delivery condition. For example, a practical issue is to predict the required time to finish a transfer given the data size, the start time and the source and destination addresses. For network designers, accurate prediction will help prepare the future needs and match the network requirements and the bandwidth in long run. For example, if a selected path is predicted to have frequent congestions, then the designer can accordingly expand the network bandwidth to meet the size of the transfers in the path or rerouting partial data flow in alternative paths.

The motivation for modeling the NetFlow measurements using Generalized Linear Mixed Model (GLMM) comes from two perspectives: (1) the features of NetFlow data; and (2) the capability of GLMM. NetFlow records are composed of multiple time series with uneven collecting time stamps, and thus the traditional single time series model is infeasible to model the data. Also, NetFlow records show mixed effects in the data, and simple consideration of fixed effects will cause information loss in the modeling [9]. Moreover, the large volume and high dimension of NetFlow data require a fast algorithm so that future transfer planning can respond quickly to the predicted future network condition. On the other hand, GLMM has the capability of flexible structure of the model in the link function, variance sourcing and incorporating mixed effect without restriction on the size of data, and thus fits our needs of analyzing the NetFlow data.

Previous research in Jiang et al. [4] and Bondell [5] discussed two issues in Linear Mixed Model (LMM), which is the GLMM with identity link and Gaussian assumption. Jiang et al. [4] shows how to obtain the best prediction in the LMM (Linear Mixed Model) and Bondell [5] uses the Lasso to select random effect in LMM for the estimation purpose. However, there are no existing methods for selecting both random effects and fixed effects for the purpose of best prediction via the Lasso [3] in LMM.

To match GLMM with the prediction interest and NetFlow data, we improve the GLMM in these aspects, in Section 2. We discuss the approach to obtain the estimates of fixed and random effects by Lasso with minimum

mean squared prediction error (MSPE) in LMM. Then, we extend the results to GLMM with the log link and Poisson assumption.

A major computational advantage of our method is that, unlike Bondell [5] and Ibrahim [10], which require to utilize the EM algorithm [11] in order to handle the unobserved random effects, our procedure does not requires the EM, then thus saves computing time.

We propose a new approach based on bootstrapping to select the optimal penalty parameter λ in Lasso. In the theoretical derivation, two advantages of this approach are analyzed: immunity to model misspecification and fast computational algorithm. After discussing the methodology in Section 2, Section 3 and Section 4 show the NetFlow data study and the simulation results, followed by the summary and discussion.

2. Methodology. In this section, we first discuss the NetFlow measurements with its format and how it matches with GLMM. Then, we show the derivation of the best predictive GLMM with Lasso.

2.1. *NetFlow dataset.* NetFlow measurements provide high volume with abundant specific information for each data flow as shown in Table 1 (with IP address is masked for privacy issues). For each record, it has the following *variables* list.

Start, End The start and end time of the recorded data transfer.
Sif, Dif The source and destination interface assigned automatically for the transfer.
SrcIPaddress, DstIPaddress The source and destination IP addresses of the transfer.
SrcP, DstP The source and destination Port chosen based on the transfer type such as email, FTP, SSH, etc.
P The protocol chosen based on the general transfer type such as TCP, UDP, etc..
Fl The flags measured the transfer error caused by the congestion in the network
Pkts The number of packets of the recorded data transfer.
Octets The Octets measures the size of the transfer in bytes.

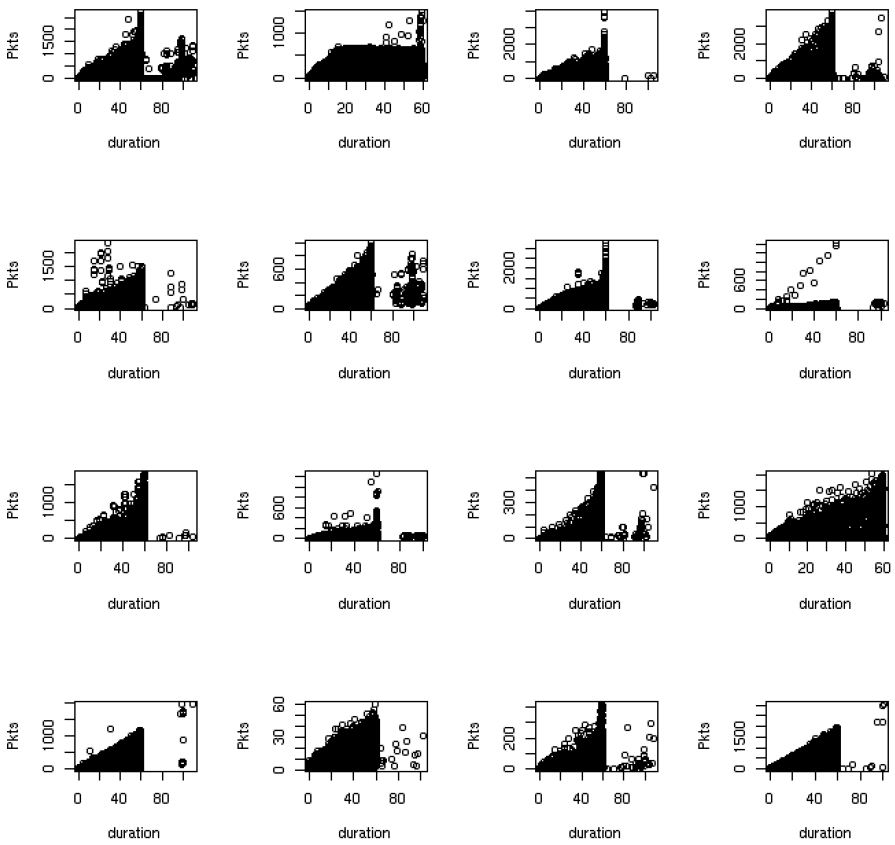
Considering features of NetFlow data and the application interests, Generalized Linear Mixed Model (GLMM) is suggested in this paper to predict the network performance for the following reasons.

- NetFlow record is composed of multiple time series with uneven collecting time stamps. Because of this feature, traditional time series

TABLE 1
NetFlow Records

Start DstIPaddress(masked)	End DstP	Sif P	SrcIPaddress(masked) Fl	SrcP Pkts	Dif Octets
0930.23:59:37.920	0930.23:59:37.925	179	xxx.xxx.xxx.xxx	62362	175
xxx.xxx.xxx.xxx	22364	6	0	1	52
0930.23:59:38.345	0930.23:59:39.051	179	xxx.xxx.xxx.xxx	62362	175
xxx.xxx.xxx.xxx	28335	6	0	4	208
1001.00:00:00.372	1001.00:00:00.372	179	xxx.xxx.xxx.xxx	62362	175
xxx.xxx.xxx.xxx	20492	6	0	2	104
0930.23:59:59.443	0930.23:59:59.443	179	xxx.xxx.xxx.xxx	62362	175
xxx.xxx.xxx.xxx	26649	6	0	1	52
1001.00:00:00.372	1001.00:00:00.372	179	xxx.xxx.xxx.xxx	62362	175
xxx.xxx.xxx.xxx	26915	6	0	1	52
1001.00:00:00.372	1001.00:00:00.372	179	xxx.xxx.xxx.xxx	62362	175
xxx.xxx.xxx.xxx	20886	6	0	2	104

FIG 1. *Relationship (Number of Packets v.s. Duration) on 16 Different Paths, showing mixed effect such as transfer path exists*



methods such as ARIMA model, wavelet analysis, and exponential smoothing model are not applicable since they are designed for evenly collected time stamps and mainly dealing with a single time series. Some researches have extended their usage in two time series, but the complexity and inefficiency block them to go beyond. Thus, there is a need for a model for multiple time series without constraints of even collection of time stamps. At the same time, GLMM can fully utilize all variables in the dataset with no need for an even-spaced time variable.

- NetFlow record is a multivariate dataset showing mixed effects. In Figure 1, we see that with the increasing number of packets in a data transfers, it takes longer time in general to finish data transfer. This suggests that the number of packets can be a fixed effect to predict the duration of a data transfer. Moreover, we see in different network paths, the fluctuation patterns in terms of slope rate and spread range are different for duration against the number of packets. This suggests that the network path for data transfer can be considered as a random effect to explain the duration under varying conditions. Thus, in terms of modeling mixed effects, GLMM has the strength over Generalized Linear Model (GLM) that only considers fixed effect, and it has the flexibility over Linear Mixed Model(LMM) that can only model continuous response variable along with Gaussian assumption. GLMM is general in a sense that it expands the choice of underlying distribution by relating the linear model to the response variable via a link function and categorizes the variance source by measuring the random effects.
- NetFlow measurements are big data with millions of observation for a single router within a day and 14 variables in each record with 30s or 40s interaction terms as candidates. The large volume of the data requires an efficient modeling. When identifying distinctive patterns within each group, traditional hierarchical models divide the data according to groups, and then model each group. However, there are three main reasons that these models are not feasible in this case:
  - The grouping factor is not clear and requires investigation to identify the variable that classify the observed data. Data explorative analysis shows that the grouping factor can be a path of the data transfer, the delivering time of the day, the transfer protocol used and their combination. If using hierarchical model, we need to model the data several times by choosing different grouping factor. However, when using GLMM, the significance of random effect will suggest the grouping factor on the response variable.

- Hierarchical models make the prediction accuracy worse, since their errors have the lower convergence rate to 0 than GLMM.
- GLMM provides one model for all dataset, while hierarchical models generate one model per each group, thus complex and inefficient.

2.2. *Generalized Linear Mixed Model.* The GLMM is defined with a vector of random effects v and the responses $y_1, ..., y_m$ of m groups that are conditionally independent such that the probability density function(pdf) of each response $f_i(y_i|v)$ follows the exponential family with

$$E(y_i|v) = \mu_i, g(\mu_i) = x_i'\beta + z_i'v, g^{-1} = h \tag{2.1}$$

where $v \sim N(0, \Phi)$, x_i is the observed fixed effect, and z_i is the index that indicates the group of random effect.

The $g(.)$ is the link function, and takes various forms such as Gaussian, Poisson and Logit with different assumptions of the model. In the NetFlow data, the only two types of varaibles are (1) continuous variables such as the size of the data transfer and the duration measured in milliseconds, and (2) count variables such as the number of congestions or the number of extreme large data transfers within a certain time window length. In order to predict these two types of response variable, the GLMM are constructed in the following two types.

- y_i is the continuous variable, and assumed $g(x) = x$, $y_i|v$ follows Gaussian distribution.
- y_i is the count variable, assumed $g(x) = log(x)$, $y_i|v$ follows Poisson distribution.

The mixed effects θ of prediction interest and its Best Predictor (BP) $\check{\theta}$ under assumed model M are

$$\theta = h(F'x\beta + R'v), \text{ where } F \text{ and } R \text{ are known matrices} \tag{2.2}$$

and

$$\check{\theta} = E_{M,\psi}(\theta_i|y) = h_{M,i}(\psi, y_i), \text{ where the parameter set } \psi = \{\beta, \Phi\} \tag{2.3}$$

M stands for the assumed model, and $h_{M,i}$ is the function showing the BP of θ connected with ψ and y_i. The MSPE to be minimized can be expressed

as

$$
\begin{aligned}
MSPE(\check{\theta}) &= E(|\check{\theta}-\theta|^2) \\
&= \sum_{i=1}^{m} E\{h_{M,i}(\psi, y_i) - \theta_i\}^2 \\
&= E\{\sum_{i=1}^{m} h_{M,i}^2(\psi, y_i)\} - 2\sum_{i=1}^{m} E\{h_{M,i}(\psi, y_i)\theta_i\} + \sum_{i=1}^{m} E(\theta_i^2) \\
&= I_1 + 2I_2 + I_3
\end{aligned} \tag{2.4}
$$

Note that, unlike $E_{M,\psi}$, which depends on the assumed model as well as the parameter ψ, the E in 2.4 is with respect to the true underlying distribution of y and θ, which may be unknown but not model dependent. This is a key feature of the approach (Jiang et.al [4]).

First consider the single case that Φ i sknown. Denote the MSPE in (2.4) by $MSPE(\beta)$. Then, it is straightforward to apply the Lasso to select the fixed effects, that is,

$$\check{\beta} = argmin_{\beta}(MSPE(\beta) + \lambda|\beta|) \tag{2.5}$$

However, selecting the random effects using the Lasso is not as simple. This is because the insignificant fixed effects are eliminated with its coefficient β diminishing exactly to 0; however, insignificant random effects are eliminated with the corresponding whole columns and whole rows of the covariance matrix diminishing exactly to 0 (Bondell [5] and Ibrahim [10]). After the covariance matrix is reformed, its positive definite property should also be maintained. In order to solve this difficulty, we use theCholesky decomposition on the covariance matrix.

$$\Phi = D\Lambda\Lambda' D \tag{2.6}$$

where D is diagonal matrix $D = diag(d_1, d_2, ..., d_q)$, and Λ is the lower triangular matrix with 1's on the diagonal.
The standardized model with the identity link function is

$$y_i = X_i\beta + Z_i D\Lambda v_i + \epsilon_i \text{ where } v_i \sim N(0, I) \tag{2.7}$$

Where $\epsilon \sim N(0, \Sigma)$. The shrinkage penalty is imposed on d_i, the element of diagonal matrix D. When d_i is shrunk to 0, the corresponding random effect is eliminated. The covariance matrix $\Phi = D\Gamma\Gamma' D$ is still guaranteed to be positive definite. After the decomposition, the original random effects coefficients Z_i^*, Z^* change into

$$Z_i = Z_i^* D\Lambda, Z = Z^*\tilde{D}\tilde{\Lambda}, \tilde{D} = I_m \otimes D, \tilde{\Lambda} = I_m \otimes \Lambda$$

The parameter set of prediction interest $\psi^* = \{\beta, \Phi\}$ changes into $\psi = \{\beta, d\}$.

2.3. *Case 1: Gaussian distribution.* LMM is a special case of the GLMM when the link function is identity, that is, $h(\mu) = \mu$ in (2.1), and the underlying exponential family is Gaussian. The mixed effect of prediction interest and its BP under assumed model M are

$$\theta = F'x\beta + R'v, \text{ where } F \text{ and } R \text{ are known matrices} \tag{2.8}$$

$$\check{\theta} = E_M(\theta|y) = F'X\beta + R'E_M(v|y) = F'X\beta + R'Z'V^{-1}(y - X\beta). \tag{2.9}$$

where $V = var(y) = \Sigma + ZZ'$

For fixed effects without Lasso, in Jiang et al [4] it shows:
With previous mentioned notation R, Z, V, and F, now write $B = R'Z'V^{-1}, \Gamma = F' - B$ and $H = Z'F - R$,

$$\begin{aligned} MSPE(\check{\theta}) &= E(|\check{\theta} - \theta|^2) \\ &= E(|H'v + F'e|^2) - 2E((v'H + e'F)\Gamma(y - X\beta)) \\ &\quad + E((y - X\beta)'\Gamma'\Gamma(y - X\beta)) \\ &= I_1 - 2I_2 + I_3 \end{aligned} \tag{2.10}$$

Since the true model tells $y = \mu + Zv + e$, so

$$\begin{aligned} I_2 &= -2E((v'H + e'F)\Gamma(y - X\beta)) \\ &= -2E((v'H + e'F)\Gamma(y - \mu)) - 2E((v'H + e'F)\Gamma(\mu - X\beta)) \\ &= -2E((v'H + e'F)\Gamma(Zv + e)) \end{aligned} \tag{2.11}$$

Among three components, I_1 and I_2 is not related to β. Since β is the only parameter that matters in the minimization of $MSPE(\check{\theta})$, the minimization is equivalent to

$$\check{\beta} = argmin((y - X\beta)'\Gamma'\Gamma(y - X\beta))$$

It's important to note that 1) the MSPE is calculated with $E(.)$, 2) the expectation under the true model rather than $E_M(.)$, and 3) the expectation related with the assumed model M. This MSPE calculation feature of this method guarantees that $\check{\beta}$ is immune to model misspecification, and proves to have better prediction accuracy than the estimates BLUP resulted from MLE. Besides the immunity to model misspecification, selecting the fixed effects via Lasso for efficient model selection is also imposed.

$$\check{\beta} = argmin_\beta(y - X\beta)'\Gamma'\Gamma(y - X\beta) + \lambda \sum |\beta_i| \tag{2.12}$$

The selection of random effects are not addressed in the Jiang et.al [4] and is now discussed in this paper. For random effects selection, the MSPE is minimized only with the part related to the parameter of interest d.

$$\begin{aligned} MSPE(\check{\theta}) = C &+ tr((Z'FF'Z - Z'FBZ - RF'Z + RBZ)G) \\ &- tr(FB\Sigma) + E((y - X\beta)'M'M(y - X\beta)) \\ &+ tr((Z'FF'Z - Z'FR' - RF'Z)G) \end{aligned} \tag{2.13}$$

where $C = 2tr(FF'\Sigma) + tr(RR'G)$ is not related to d
Applying the L_1 penalty along with MSPE to achieve the efficiency model selection with Lasso, the random effects are

$$\begin{aligned} \check{d} = &argmin_d(y - X\beta)'M'M(y - X\beta) \\ &+ tr((2HBZ - HF'Z - Z'FR')G) - tr(FB\Sigma) \\ &+ \lambda \sum |d_i| \end{aligned} \tag{2.14}$$

The final model for prediction is built with equation (2.12) and (2.14) and has two distinctive advantages. First, it is mentioned in the paper that the minimization problem is not based on the assumed model M and thus immune to the model misspecification errors. Second, the computation complexity is only a minimization problem with $O(nP)$ where n is the number of observation and P is the number of parameters in the full model. It is much simpler than the MCEM algorithm that is required by Bondell [5] and Ibrahim [10] in order to handle the unobserved random effects in their estimation-based penalized maximum likelihood algorithms.

2.4. *Case 2: Poisson distribution.* The second case is when the response variable is the counted data. Given the small area means $\mu_1, ..., \mu_m$, the observations $y_1, ..., y_m$ (with y_i being from the ith small area) are independent such that

$$y_i \sim Poisson(\mu_i); log(\mu_i) = x_i'\beta + z_i v_i \tag{2.15}$$

The vector of prediction interest and its BP is

$$\theta = h(F'x\beta + R'v), \check{\theta} = E_{M,\psi}(\theta_i|y) = h_{M,i}(\psi, y_i) \tag{2.16}$$

Utilizing the properties of Poisson distribution as derived in Appendix A,

$$\begin{aligned} MSPE(\check{\theta}) &= E\{\sum_{i=1}^{m} h_{M,i}^2(\psi, y_i)\} - 2\sum_{i=1}^{m} E\{h_{M,i}(\psi, y_i)\theta_i\} + \sum_{i=1}^{m} E(\theta_i^2) \\ &= E\{\sum_{i=1}^{m} h_{M,i}^2(\psi, y_i) - 2\sum_{i=1}^{m} h_{M,i}(\psi, y_i - 1)y_i + \sum_{i=1}^{m} E(\theta_i^2)\} \end{aligned} \tag{2.17}$$

Since $\sum_{i=1}^{m} E(\theta_i^2)$ has no relationship with the parameter set $\phi = \{\beta, d\}$, minimizing MSPE is equivalent to minimizing.

$$Q(\psi) = \sum_{i=1}^{m} h_{M,i}^2(\psi, y_i) - 2\sum_{i=1}^{m} h_{M,i}(\psi, y_i - 1)y_i \tag{2.18}$$

The fixed and random effects under Poisson cases are

$$\beta = argmin_\beta Q(\psi) + \lambda_\beta \sum_{j=1}^{p} |\beta_j|, \quad d = argmin_d Q(\psi) + \lambda_d \sum_{i=1}^{m} |d_i|. \tag{2.19}$$

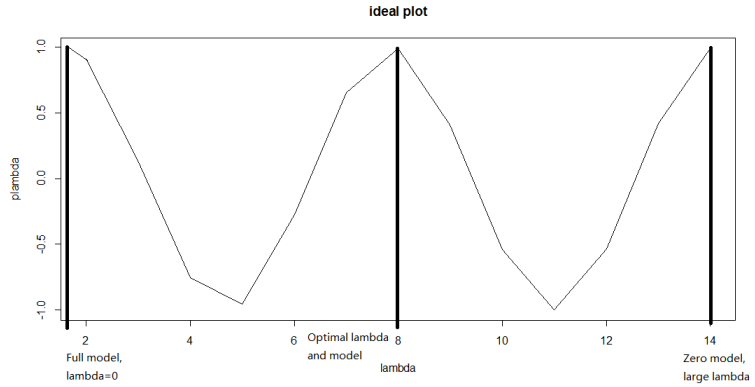
2.5. *Selecting Penalty Parameters in Lasso.* In Jiang et. al [6], the adaptive fence procedure raises a method to select the tuning constant c_n that is used in the model selection. The idea is that it is ideal if the selecting tuning constant c_n maximizes the probability of choosing the optimal model. Suppose M is the set of candidate models which includes the optimal model M_{opt}, and the selected model is $M_0(c_n) \in M$. Then, optimal c_n should be

$$c_n = argmax_{c_n} P(M_0(c_n) = M_{opt}) \tag{2.20}$$

In order to find c_n through formula (2.20), two keys must be known: (1) the underlying distribution to compute P; and (2) M_{opt}.
In Jiang et. al [6], the first key, probability distribution P can be approximated by the bootstrapped samples under the full model M_f. The second key, M_{opt} can be found utilizing the idea of maximum likelihood. The optimal model is the model that generates data, and thus should be the model that is favored the most by the data. Since the bootstrapped samples almost duplicate the information from original data, M_{opt} is the most supported by the bootstrapped samples, i.e. most frequent to be selected. Extending the adaptive fence idea into the penalty parameter selection λ_n in Lasso, the procedures are:

- Step 1: Fit the full model M_f and bootstrap B samples from it
- Step 2: Select a grid of λ. For each λ, record $M_*(\lambda)$, the model that is selected the most, across B samples, i.e. $M_*(\lambda) = argmax_M P(M_0(\lambda) = M(\lambda))$ where $P(M_0(\lambda) = M(\lambda))$ is counted as the portion that the number of samples support model $M(\lambda)$ as the selected model $M_0(\lambda)$ out of B.
  Note here that the final selected model $M_*(\lambda)$ is related to λ. When $\lambda \to 0$, the model that is selected the most $M_*(\lambda)$ will be the full model M_f, and when $\lambda \to 0$, the model that is selected the most $M_*(\lambda)$ will be the empty model M_{empty}.

FIG 2. *The Ideal(with unique peak in the middle) Plot for Selecting Tuning Parameter* λ



- Step 3: Denote $P_*(\lambda) = P(M_0(\lambda) = M_*(\lambda))$ as the support percentage of the favorite $M_*(\lambda)$ given λ. Plot $P_*(\lambda)$ against λ. And in the ideal plot as Figure 2 with a unique peak in the middle, this peak is the favorite tuning parameter λ.

In Figure 2, the two ends have peaks because when λ is either too small or too large, only the full model M_f and the empty model M_{empty} will be selected. In Jiang's approach [6], the λ corresponding to the peak in the middle of the plot should be the chosen λ, which maximizes the probability that the selected model is equal to the optimal model, $M_0(c_n) = M_{opt}$. The ideal situation does not always show, and in many times, one will end up in either of the cases shown in Figure 3. The fluctuation in the left case occurs due to the variation from the observed data and bootstraps. The platform in the right case occurs due to the fine cut in the grid of λ.

To solve these two problems, one no longer uses evenly-spaced grid of λ, but a dimension-related λ that λ_j corresponds to the j-predictors model. The detailed new approach goes through the following steps:

- Step 1: Start from smallest $\lambda_p = 0$. It returns M_p, a full model with p predictors. Keep increasing λ, until M_{p-1}, a model with $p-1$ predictors, is returned. Record the current value as λ_{p-1}
  $[\lambda_p, \lambda_{p-1})$ is the range that model with p predictors are chosen.
- Step 2: Keep increasing λ, until one gets all the ranges for the dimension wise model selection. For $i = 0, ..., p, [\lambda_i, \lambda_{i-1})$ is the range that models with i predictors are chosen.
- Step 3: For each range $[\lambda_i, \lambda_{i-1})$, evenly separate the range into a grid by k candidate λs. For each λ, compute the model across all bootstrap

FIG 3. *The Common (without unique peak in the middle) Wiggling and Platform Plots for Selecting Tuning Parameter* λ

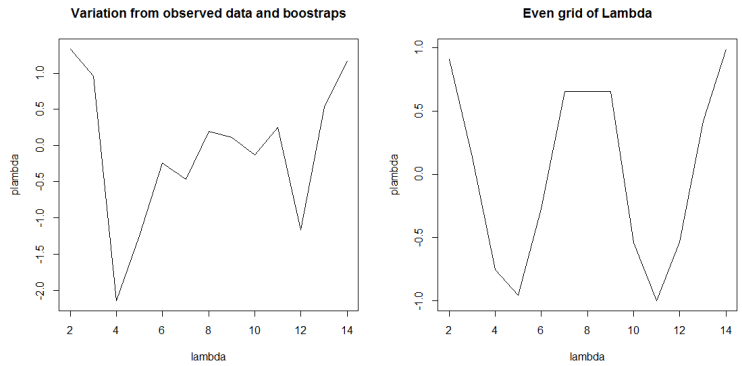


TABLE 2
The New Approach to Solve Wiggling and Platform: Dimensional Selection for Tuning Parameter λ

dim	p	p-1	...	i	...	0
λ range	$[\lambda_p, \lambda_{p-1})$	$[\lambda_{p-1}, \lambda_{p-2})$	...	$[\lambda_i, \lambda_{i-1})$	...	$[\lambda_0, \lambda_{-1})$
λ_i^*	λ_p^*	λ_{p-1}^*	...	λ_i^*	...	λ_0^*
p_i^*	p_p^*	p_{p-1}^*	...	p_i^*	...	p_0^*

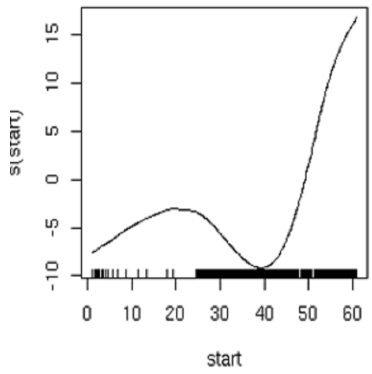
samples, and chose the optimal λ within this range as λ_i^* and the corresponding $p_i^*(M_i^*, \lambda_i^*)$. Table 2 summaries the tuning parameters' range, the best λ and its supported percentage for each dimension.

- Step 4: Plot p_i^* against λ_i^* that are summarized in Table 2. The middle peak is selected as the overall optimal λ^*, and its corresponding model is selected as the final optimal model M^*.

The platform case is solved because each λ now selects the model with different number of predictors, and the corresponding probability is not likely to be the same in the neighboring range. The variation case is solved because more robust and sophisticated choice of λ_j eliminates the unwanted wiggling in the plot. The resulted plot is more close to the shape in the ideal case shown in Fig. 2.

3. NetFlow Data Study. The sample NetFlow data is provided by ESnet for the duration from May 1 2013 to June 30 2013. Considering the network users' interests, the established model should predict the duration of a data transfer so that users will expect how long the data transfer would take, given the size of their data, the start time of the transfer, selected path and protocols. Considering the network designers' interests, the established

FIG 4. *Smoothing Spline Transformed Start Time Variable, showing the nonlinear relationship between start time and transfer duration*



model should predict the long time usage of the network, so that the designer will know which link in the network is usually congested and requires more bandwidth or rerouting the path. In the following, the models for these two interests are built, and its prediction accuracy is compared with two traditional GLMM algorithms: Backward-Forward selection and Estimation-based Lasso.

3.1. *GLMM on Duration.* The full model predicts the transfer duration, assuming influences from the fixed effects including transfer start time, transfer size (Octets and Packets) and the random effects including network transfer condition such as Flag and Protocol, source and destination Port numbers and transfer path such as source and destination IP addresses and source and destination Interfaces. After selecting and fitting the model via our Predictive Lasso procedure, the final model is

$$y = \beta_{start} s(x_{start}) + \beta_{pkt} x_{pkt} + Z_{ip-path} v_{ip-path} + e \tag{3.1}$$

Since time variable is usually not linear related with the response variable, smoothing spline transformation $s(.)$ is implemented to the time variable and the smoothing parameters are chosen automatically by cross-validation. P-value in the final model is all less than 2e-16, which stands for the significance of those variables in the model. The significant fixed effects in the model are start time and number of packets, as shown in Table 3. The transformed start time data is plotted in Figure 4 and the plot tells how the duration varies for different start time.

FIG 5. *Standard Deviation Estimates for Random Effects in GLMM (3.1) to Predict The Transfer Duration, showing the busier paths bring higher variation to the transfer duration*

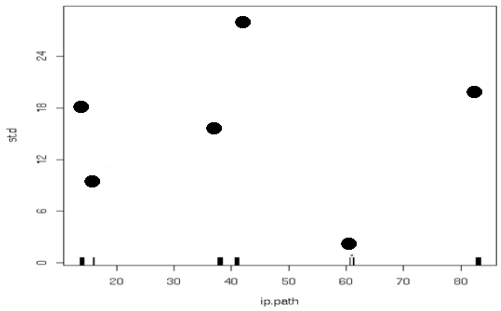


TABLE 3
Coefficient Estimates for Fixed Effects in GLMM (3.1) to Predict the Transfer Duration

Fixed Effects	Estimates	Standard Deviation	P-value
Intercept	-13.809	0.914	<2e-16
Start Time	0.574	0.0169	<2e-16
Packets	1.115	0.035	<2e-16

TABLE 4
Comparison of MSPE and Speed to Predict Duration

	Est.Lasso s	BF Selection	Pred. Lasso
MSPE	2306	42230	127.3
Modeling Time (in seconds)	6.26e+7	5.43e+10	142

The random effects' standard deviance estimates are plotted in Figure 5, and the plot shows the traffic duration varies with the different IP paths. In our sample, there are six paths indexed as 83, 38, 41, 14, 16 and 61.The index is categorical representation of the IP path and has no numerical values in the model. The busier paths, such as paths 83, 38, 41 and 14 that have dense area shown in the bottom of Figure 5 come with higher fluctuation rates in the transfer duration. While paths 16 and 61 have less traffic and lower variation rates, besides the uncertainty resulted from each IP path, the background noise is estimated with a standard deviation of 11.2392.

The model suggests the importance of variation in random effects such as IP path in the prediction of the duration. Besides the path, start time selection and assignment of packets are also significant in the prediction of the duration. Compared to the other two approaches in Table 4, the Predictive Lasso shows that the best prediction accuracy is 18 times better than the Estimation Lasso and 330 times better than the Backward-Forward Selection and the least computation time is 4e+5 times less than the Estimation Lasso and 3.8e+8 times less than the Backward-Forward Selection. The Predictive Lasso greatly improves the prediction accuracy which fits the interests of modeling and also provides efficient fast algorithm compared to the Estimation Lasso and Backward-Forward Selection, as analyzed in section 2.

3.2. *GLMM on Frequency of Congestion.* This model predicts the frequency of congestion occurred in each link of the network. The response variable y is the number of congestion measured by the speed, BytesPerSecs. A congestion event is defined when BytesPerSecs is less than 50, which is the slowest 10% of network transfer speed. The full model to predict number of congestion assuming influence from two sources: fixed effects and random effects. The fixed effects includes transfer size (Octets and Packets), number of transfers with their Protocol is 6, 17, 47 and 50 respectively and number of transfers with their Flag is 0,1,2 and 4 respectively. And random effect is transfer path, the source and destination IP address. After selecting and fitting the model via our Predictive Lasso procedure, the final model is

$$logE(y|v) = \beta_{pkts}x_{pkts} + \sum \beta_{p=i}x_{p_i} + Z_{ip-path}v_{ip-path} \tag{3.2}$$

The significant fixed effects in the selected model are the transfer size, number of packets and the protocol used, as shown in Table 5.

The random effects' standard deviance estimates are plotted in Figure 6, and the plot shows that the traffic duration in Y axis varies in different transfer IP paths in X axis. In our sample, there are 414 paths, and the busier path comes with the higher fluctuation rates. Besides the uncer-

TABLE 5
Coefficient Estimation for Fixed Effects in GLMM (3.2) to Predict The Frequency of Congestion

Fixed Effects	Estimates	Std	P-value
Intercept	816.95627	0.78305	<2e-16
Packets	28.53924	0.02284	<2e-16
Protocol=6	45.43606	0.01608	<2e-16
Protocol=17	-1.58644	0.14088	<2e-16
Protocol=47	-8.39576	0.36338	<2e-16
Protocol=50	-4.96028	0.05175	<2e-16

FIG 6. *Standard Deviation Estimates for Random Effects in GLMM (3.2) to Predict The Frequency of Congestion, showing the busier paths bring higher variation to the frequency of congestion*

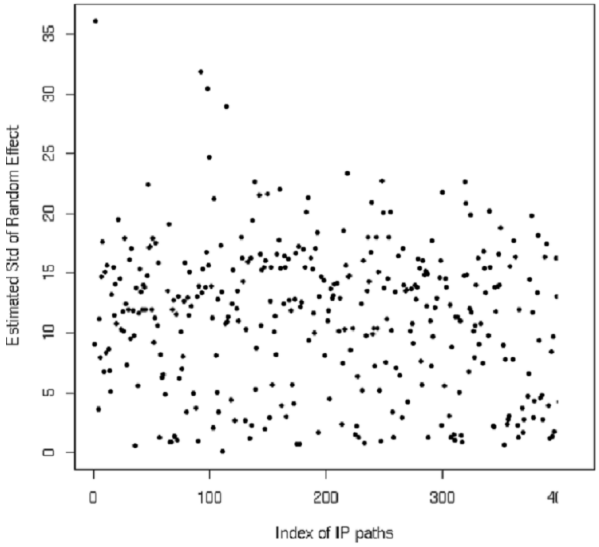


TABLE 6
Comparison of MSPE and Speed to Predict Counts of Congestion

	Est.Lasso s	BF Selection	Pred. Lasso
MSPE	27.7	42.5	12.73
Modeling Time (in seconds)	7.31e+8	1.24e+10	10.06e+2

tainty resulted from each IP path, the background noise is estimated with a standard deviation of 25.316.

The model suggests the importance of variation in random effects such as IP path in predicting the congestion frequency. Besides the path, the protocol selection and assignment of packets also significantly affect the congestion rate. Compared to the other two approaches in Table 6, the Predictive Lasso shows the best prediction accuracy which is twice better than the Estimation Lasso and four times better than the Backward-Forward Selection, and the least computation time in modeling step which saves 7e+8 seconds than the Estimation Lasso and 1e+10 seconds than the Backward-Forward Selection. Although the prediction accuracy improvement this case by the Predictive Lasso is not as dramatically as in the previous case, the saving in computing time is even much more impressive.

4. Simulation. The Predictive Lasso developed in section 2 is shown in section 3 to have two main advantages in terms of better prediction accuracy and less computational cost than the estimation-oriented methods. In this section, we use simulation studies to furthur support and illustrate these two advantages. The comparison is among the Estimation Lasso, the Backward-Forward Selection and the Predictive Lasso. The first advantage of better prediction accuracy is due to two reasons: First, the optimization is calculated without using E_M or the distribution of the assumed model. Thus the parameter estimates are not affected by model misspecification error. Secondly, the Predictive Lasso minimize the MSPE, while the estimation-oriented methods target to maximize likelihood. In this way, the Predictive Lasso gets smaller prediction error both for fixed effects prediction and random effects prediction. Here three scenarios are considered: increasing the variance, increasing number of observations in each group and increasing number of groups.

The simulation data is generated from the true model,

$$M = \beta_0 + \beta_1 x_{ij1} + \beta_2 x_{ij2} + z_{ij1} v_{1i} + z_{ij2} v_{2i}$$

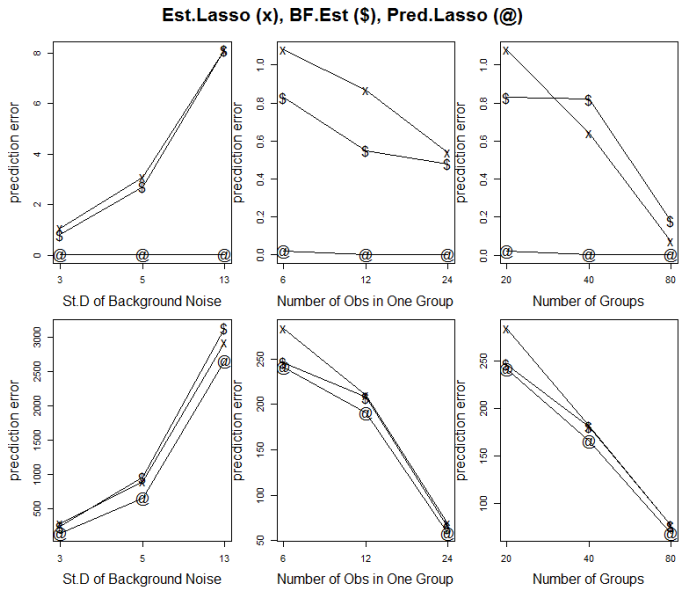
$$i = 1, ..., N; j = 1, ...n_i; \phi = diag(sd_1^2, sd_2^2); sd_1 = 3; sd_2 = 2$$

$$N = 20; n_i = 6; var(e_{ij} = sd^2 = 3)$$

The Gaussian model is given as $\mu = M$, and the Poisson model is given as $log(\mu) = M$. However, combined with redundant observed information and model misspecification, the assumed model is

$$\begin{aligned} M = {} & \beta_0 + \beta_1 x_{ij1} + \beta_2 x_{ij2} + \beta_3 x_{ij3} + \beta_4 x_{ij4} + \beta_5 x_{ij5} + \beta_6 x_{ij6} \\ & + z_{ij1} v_{1i} + z_{ij2} v_{2i} + z_{ij3} v_{3i} + z_{ij4} v_{4i} + z_{ij5} v_{5i} + z_{ij6} v_{6i} + z_{ij7} v_{7i} \end{aligned} \tag{4.1}$$

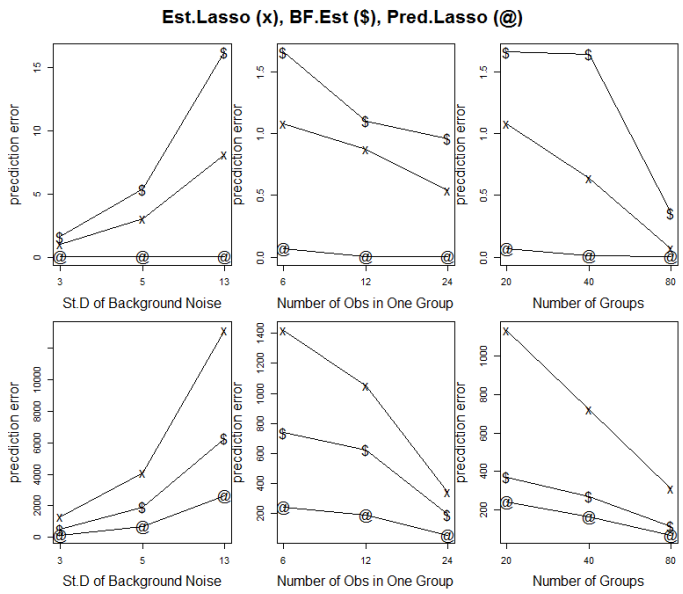
FIG 7. *Prediction Accuracy under Case 1: Gaussian Model, showing the Predictive Lasso has the smallest prediction error*



The assumed model is misspecified altogether 4 redundant fixed effects and 5 redundant random effects. The simulation is carried out for the two types of GLMM and under three scenarios: increasing variance (sd), increasing number of observations in each groups n_i, and increasing number of groups N. From Figure 7 showing the Gaussian Model and Figure 8 showing the Poisson Model, the plots on the first row show the performance of fixed effect prediction accuracy, and the plots on the second row show the performance of random effect prediction accuracy. The plots on the left column, regarding increasing variance, show that the Predictive Lasso does not significantly worse in terms of prediction error than the other two methods. The plots on the middle column, regarding increasing number of observations in each groups, and the plots on the right column, regarding the increasing number of groups, both show that the Predictive Lasso always holds the most accuracy position no matter how the data is segmented into groups.

The second advantage of Predictive Lasso is the dramatically reduced computational costs in reaching the final model. In the optimization steps, the Estimation Lasso and the Backward-Forward Selection require MCEM to estimate the expectation of the likelihood of the assumed model, since their target function involves non-observed random effects. However, the

FIG 8. *Prediction Accuracy under Case 2: Poisson Model, showing the Predictive Lasso has the smallest prediction error*

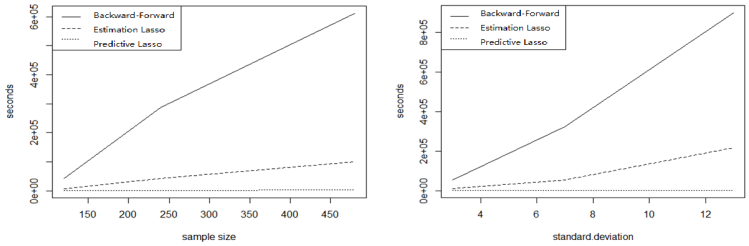


Predictive Lasso has an optimization function without the unobserved random effects which eliminate the costly MCEM. The computational complexity of the optimization problem in Predictive Lasso is $O(np)$, where n is the observation and p is the number of predictors, as in formula 4.2. The Estimation Lasso requires MCEM, an iterative algorithm that each iteration contains optimization and requires several iteration steps until it reaches the convengency in the final model. The Backward-Forward Selection requires l steps to reach the final model, and in each step it needs trial-and-error to decide which variable to drop or add after using MCEM to fit each candidate model as shown in formula 4.4.

$$\begin{aligned} Time(PredictiveLasso) &= Optimization \times 1; \\ Complexity(PredictiveLasso) &= O(np) \end{aligned} \tag{4.2}$$

$$\begin{aligned} Time(EstimationLasso) &= MCEM \times k \\ &= (MC + Optimization) \times k; \\ Complexity(EstimationLasso) &= O(npk) \end{aligned} \tag{4.3}$$

FIG 9. *Computational Costs of Three Methods, showing the Predictive Lasso has the least computation time for both the size of the data increases (left) and the uncertainty in the data increases (right)*



where k is the number of iteration before algorithm converges

$$\begin{aligned} Time(BF\ Selection) &= MCEM \times \sum_{i=1}^{l} k_i \sum_{j=1}^{J_i} n_{ij}; \\ Complexity(BF\ Selection) &= O(np \sum_{i=1}^{l} k_i \sum_{j=1}^{J_i} n_{ij}) \end{aligned} \tag{4.4}$$

where l is the number of steps before reaching to the final model; k_i is the trial-and-error in the i-th step before moving to the $i+1$-th step; J_i is the number of remaining variables in the model at i-th step, and n_{ij} is the number of iteration before MCEM converges for the i-th trial with j-th predictors omitted.

From the equation 4.2, 4.3 and 4.4, it is clear that the Predictive Lasso saves large computational costs compared to the Estimation Lasso and the Backward-Forward Selection. Moreover, the simulation supports the computational advantage of the Predictive Lasso. In Figure 9, the left graph shows the increasing computational time when the sample size is increased, and the right graph shows the increasing computational time, when the variation of data is increased. The Predictive Lasso costs the least time in computation, when the data volume is increased and uncertainty in the data is increased. This feature perfectly meets the need of large volume and high fluctuation from the NetFlow measurements.

The simulation examines the two advantages of the Predictive Lasso. Firstly, under Gaussian model and Poisson model, the results show that the Predictive Lasso has much smaller prediction error than the Estimation Lasso and the Backward-Forward Selection. Secondly, the computational complexity listed in the formula and the simulation result both show that the magnitude of the consumed computational time by the Predictive Lasso

is many times less than the other two methods.

5. Summary of Discoveries and Discussion. Large scientific data movements require efficient utilization of the network bandwidth. Network performance prediction helps scheduling and estimation of the large network usage. Some of challenges in the prediction with large data movement are the computational cost and the large number of features in the data. The conventional methods such as Estimation Lasso and the Backward-Forward Selection are very computationally costly. Computational complexity is $O(npk)$ for Estimation Lasso shown in (4.3) and $O(np\times\sum_{i=1}^{l} k_i \sum_{j=1}^{J_i} n_{ij}$ for Backward-Forward Selection shown in (4.4), thus may not handle large data set with n observations and p dimensions. To solve this problem, we developed an efficient statistical method, the Predictive Lasso which finishes the prediction task without multiple iterations. The computational complexity of the proposed Predictive Lasso is only O(np) shown in (4.2), thus can handle the large volume of data. In many cases, large data sets include multiple features. The features degenerate the input data set into numerous, smaller partitions. Handling such individual partitions become intractable when the number of features grows. To solve this issues, we propose the GLMM with Lasso model. The GLMM prevents the input data set degenerating by specifying common features.

Specifically, we presented the analysis of network measurement data to predict the network traffic for efficient utilization of the network bandwidth for large scientific data transfers as well as capacity planning of the network infrastructure up to the future bandwidth needs. Our Predictive Lasso combines the best prediction in GLMM and the efficient model selection of Lasso. The method is designed by minimizing the MSPE plus the L-1 penalty on the coefficients of fixed effects and random effects. Compared to the Estimation Lasso and Backward-Forward Selection, our method holds the best prediction accuracy and the least computational costs, supported by the simulation study and real application on the NetFlow measurement data. In addition, we developed an innovative approach for selecting tuning parameters, based on dimensional modeling with bootstrapping. The Predictive Lasso method will be used to model the performance of the data flow over to predict the network traffic bandwidth in support of efficient utilization of the network infrastructure.

APPENDIX A: DERIVIATION OF CONDITIONAL EXPECTATION UNDER POISSON CASE

Under the Poisson case of GLMM, the overall MSPE can be expressed as

$$MSPE(\check{\theta}) = E\{\sum_{i=1}^{m} h_{M,i}^2(\psi, y_i)\} - 2\sum_{i=1}^{m} E\{h_{M,i}(\psi, y_i)\theta_i\} + \sum_{i=1}^{m} E(\theta_i^2)$$

Utilizing the property of Poisson distribution, the second part of MSPE can be written as the following

$$\begin{aligned} E\{h_{M,i}(\psi, y_i)\theta_i\} &= E[\theta_i E\{h_{M,i}(\psi, y_i)|\theta\}] \\ &= \sum_{k=0}^{\infty} h_{M,i}(\psi, k) E(e^{-\theta_i}\theta_i^{k+1}/k!) \\ &= \sum_{k=0}^{\infty} h_{M,i}(\psi, k)(k+1) E\{e^{-\theta_i}\theta_i^{k+1}/(k+1)!\} \end{aligned} \tag{A.1}$$

where $\theta = (\theta_i)_{1 \le i \le m}$. Further more,

$$E\{e^{-\theta_i}\theta_i^{(k+1)}/(k+1)!\} = E\{1_{(y_i=k+1)}\}.$$

Thus, with $h_{M,i}(\psi, -1) = 0$,

$$\begin{aligned} E\{h_{M,i}(\psi, y_i)\theta_i\} &= E\{\sum_{k=0}^{\infty} h_{M,i}(\psi, k)(k+1)1_{(y_i=k+1)}\} \\ &= E\{h_{M,i}(\psi, y_i - 1)y_i\} \end{aligned} \tag{A.2}$$

And the overall MSPE is

$$MSPE(\check{\theta}) = E\{\sum_{i=1}^{m} h_{M,i}^2(\psi, y_i) - 2\sum_{i=1}^{m} h_{M,i}(\psi, y_i - 1)y_i + \sum_{i=1}^{m} E(\theta_i^2)\}$$

ACKNOWLEDGEMENTS

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. We thank Chris Tracy, Jon Dugan, Brian Tierney, Inder Monga and Gregory Bell at ESnet; Arie Shoshani, Joy Bonaguro and Jay Krous at LBNL; Richard Carlson at Dept. of Energy; and Demetris Antoniades and Constantine Dovrolis at Georgia Tech for their support.

REFERENCES

[1] Stallings, W.(1999). *SNMP, SNMthp2, SNMPv3 and RMON 1 and 2*, Addison-Wesley

[2] Cisco Systems Inc. (1966). *NetFlow Services and Applications - White paper*

[3] Tibshirani, R.(2011). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B* 58,267288.

[4] Jiang, J.,Nguyena T. and Rao,J. S.(2011). Best Predictive Small Area Estimation. *Journal of the American Statistical Association* 106:494, 732-745

[5] Bondell, H. D., Krishna, A. and Ghosh, S. K. (2010). Joint variable selection for fixed and random effects in linear mixed effect models. *Biometrics* 66, 10691077

[6] Jiang, J., Rao, J.S., Gu, Z. and Nguyena T. (2008). Fence methods for mixed model selection *Annals of Statistics* 36-4,1669-1692

[7] Hu, K., Sim, A., Antoniades, D. and Dovrolis, C.(2013). Estimating and Forecasting Network Traffic Performance Based on Statistical Patterns Observed in SNMP Data *MLDM 2013* 601-615

[8] Antoniades, D., Hu, K., Sim, A., Dovrolis, C.(2013) What SNMP Data Can Tell Us about Edge-to-Edge Network Performance *PAM 2013* 267-269

[9] Hu, K., Choi, J., Jiang, J. and Sim, A.(2013) Best Predictive GLMM using LASSO with Application on High-Speed Network *LBNL Tech Report 6327E, 2013*

[10] Ibrahim, JG., Zhu, H., Garcia, RI. and Guo R. (2011) Fixed and Random Effects Selection in Mixed Effects Models *Biometrics, 67, 495-503*

[11] Dempster, A.P., Laird, N.M. and Rubin, D.B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm *Journal of the Royal Statistical Society, 1-38*

Department of Statistics
One Shields Ave., Davis,CA,U.S.95616
E-mail: kjhu@ucdavis.edu
jiang@wald.ucdavis.edu

Scientific Data Management Research Group
Computational Research Division
Lawrence Berkeley National Laboratory
1 Cyclotron Road, Berkeley, CA, U.S. 94720
E-mail: jaesik@unist.ac.kr
asim@lbl.gov